

Patently Obvious?

Patenting software: Is it possible? Is it worth it?

By Jorge Contreras, Sarah Harrop,
and Florian van Baum

Software architecture and source code are often considered the “crown jewels” of a software company. To protect these assets, many software companies have begun to seek patent protection for their products, from desktop applications to telemetry algorithms. But is patent protection available for all software products? And if so, are patents worth the time, energy, and significant expense required to obtain them?

Patents, patents everywhere

Today, many commercial organizations employ veritable armies of patent lawyers and agents to generate patent application after patent application throughout the world. Last year, one company, IBM, was awarded more than 3,400 US patents – that’s almost 10 per day! Considering that the average time required to prosecute a patent from application to issuance is nearly two years, the internal processes involved with such an undertaking are staggering.

And why do companies like IBM spend millions to acquire vast portfolios of patents across the globe? Because patents create legal monopolies. The holder of a patent generally has a period of 20 years from the filing of the patent application during which no one else may practice the invention claimed by the patent. This is powerful protection, because it covers ideas, and gives the holder an exclusive

right to exploit the patented invention in the country where the patent has been granted.

This type of protection creates a powerful competitive advantage in fast-moving technology markets, where product life cycles can be measured in months rather than years. Particularly in the world of computer software, where technical innovation is necessary to keep ahead of the competition and functionality continuously evolves, one would think that the competitive edge conferred by a few strategic patents would be irresistible to most companies.

But a closer look at the patent leaders reveals some interesting facts. According to statistics published by the US Patent and Trademark Office (PTO), most of the enterprises obtaining 40 or more patents during 2001 were in industries such as computer hardware, electronics, telecommunications, optics, instrumentation, chemicals, energy, transportation, and pharmaceuticals. Notably absent were software companies. In fact, of the top 50 US patent awardees in 2001, only one, Microsoft, was predominantly a software company, coming in at number 41 with 396 patents awarded. No other company principally in the software business even came close to this figure.

So why are software companies the poor cousins of the patent game? A piece of code is just as “technological” as a network router or a

Different Outlooks for Software Patents

US Any new and useful process is patentable regardless of whether the subject matter is computer software. The focus should not be on the particular technology, but rather on the requirement that patentable inventions be

“new and useful.” **EU** Software programs are not patentable subject matter and the fact that a software program runs a computer is not, in itself, a technical effect. **UK** A software program may be patentable provided that the software program produces a technical effect when run on a computer.

Germany The mere fact that a software program is implemented on a computer does not necessarily lead to the conclusion that the invention is technical by its nature, and thus patentable.

laser pointer, isn’t it? Writing a complex piece of software code requires more or less the same degree of technical acumen as designing a new airfoil, doesn’t it? From a legal standpoint, the answer is “yes and no.”

The evolution of US software patents

Historically, computer software programs were not patentable. Until 1981, the PTO refused to grant patents on computer programs because they were viewed as mere “mathematical algorithms.” While patents protect novel and inventive processes, machines, articles of manufacture, and compositions of matter, they do not protect laws of nature, natural phenomenon, and abstract ideas, which can by their nature be discovered, but not invented. The law of gravity, for example, does not constitute patentable subject matter.

Software, the courts held, was more

appropriately covered by copyright law, which protects books, movies, song lyrics and other works of "authorship." Because software is written in a computer programming language, it was often analogized to a written work, and most countries have adopted statutes explicitly covering computer software in their copyright laws. The main difference between copyright protection and patent protection is that patents protect ideas, and the owner of a patent can prevent everyone else from making, using, or selling any product that embodies that idea. Copyrights, in contrast, only protect the specific "expression" of the idea. In the case of a book, for example, copyright would protect the words an author used to describe a picturesque seaside resort, but it would not prevent another author from describing the exact same resort in different words. The same held true for computer programs. Different software programs could perform identical functions so long as the programming code used to generate those functions was different. This approach continues to be applied to computer software protection throughout much of the world.

In 1981, however, the US Supreme Court opened the door to software patents in the United States. In the landmark case of *Diamond v. Diehr*, the court held that a computer program used to control heating times in a rubber curing process was patentable, as the invention was not merely a mathematical algorithm, but a process for molding rubber. The fact that a computer program executed the process did not make the invention non-patentable.

By 1996 the PTO issued formal examination guidelines for computer-related inventions, which stated that patents would be issued on computer programs embodied in tangible media and for sequences of operations executed by a computer, provided that they meet the other criteria for patent protection (for example, they are novel, useful and non-obvious).

Just four years later, in *State Street Bank and Trust v. Signature Financial Group*, the Federal Circuit set the current state of United States patent law as follows: any subject matter may be patentable – the focus should not be on the particular technology, but rather on the

Software Patents: Pros and Cons

Pros:

- The patentee gets a de facto monopoly over an idea
- Patents are recognizable, valuable assets
- Patents can be licensed for royalties or cross-licensed in exchange for the right to operate under a competitor's patents

Cons:

- The patenting process may be costly and time consuming
- The patent application is public and may be reviewed and/or challenged by third parties
- The validity of software patents continues to change in the EU and elsewhere

requirement that patentable inventions be "new and useful."

The software patent debate in the EU

It is a common misperception that patent protection for software is not available outside of the US. In fact, in European Union member states there are numerous filings for patent protection of software related inventions, particularly in the UK and Germany; however, as well as fulfilling the standard patent criteria, in order to be patentable a software program must also have a "technical effect."

Under the European Patent Convention (which has been adopted in the law of EU member states) certain inventions are not considered to be patentable subject matter. Such excluded inventions specifically include software programs. However, this exclusion applies only to the extent that the patent seeks to cover software "as such." The scope of the words "as such" is the key to the patentability of software programs in the EU. To establish whether a computer software program is patentable, the European Patent Office (EPO) has developed the concept of "technical effect": the EPO determines whether the software invention is a technical response to a technical problem. The EPO's landmark decision in the 1987 *Vicom* case helps shed some light on what the EPO means by this:

Vicom involved an application for a program that scanned images digitally and in doing so increased processor speed – there was considered to be technical effect (increase in processor speed) on a physical entity (the image, which was stored electronically).

However, the EPO has continued to struggle with the extent to which technical effect can be found in a computer "as programmed" and has had even more trouble with software programs either alone or as running on a computer.

Although there have been decisions that push the law in both directions, the general view is that the electrical manipulation of a computer by a software program is not a technical process. The proposed EU Software Patent Directive (currently in draft form) backs this up by taking the stance that software programs, either on their own or as being executed on a computer, are not patentable subject matter and the fact that a software program runs a computer is not, in itself, a technical effect.

The UK approach

While the decisions of the EPO are not binding on the UK courts, the approach of the UK courts has been to follow the approach of the EPO. Like the EPO, the UK courts have struggled with the extent to which a computer "as programmed" could have a technical effect, and thus be suitable for patent protection. The UK patent office has stated that it will accept patent applications for software programs, either themselves or on a carrier, provided that the software program produces a technical effect when run on a computer. This goes beyond what is proposed by the draft EU Software Directive and, in the future, the UK patent office is likely to bring its practice in line with the EU Software Directive.

The German approach

Germany has a similar approach to the UK with respect to granting patents for software programs. The German Federal court has in recent years loosened the restrictions on software inventions and has extended the scope of software patents that are available. For example, in the 1999 language analyser decision, the court accepted the technical character of a software program that analyzed